

QUANTIFYING PARAMETERS FOR BAYESIAN PRIOR
ASSUMPTIONS WHEN ESTIMATING THE PROBABILITY OF FAILURE
OF SOFTWARE

by
Jacquelyn E. Long
Assistant Professor
Department of Computer Science
Norfolk State University
Norfolk, Virginia 23504

Software reliability has become increasingly important, especially in life-critical situations. The ability to measure the results of testing and to quantify software reliability is needed. If this is accomplished, a certain minimum "amount" of reliability for a piece of software can be specified, and testing and/or other analysis may be done until that minimum number has been attained.

There are many models for estimating software reliability. The accuracy of these models has been challenged and many revisions for the models and recalibration techniques have been devised. Of particular interest is the method of estimating the probability of failure of software when no failures have yet occurred in its current version as described by Miller et al[3]. This model uses black box testing with formulae based on Bayesian estimation. The focus is on three interrelated issues: estimating the probability of failure when testing has revealed no errors; modifying this estimation when the input use distribution does not match the test distribution; and combining the results from random testing with other relevant information to obtain a possibly more accurate estimate of the probability of failure. My research relates directly to the third issue, obtaining relevant information about the software and combining the results for a better estimate for the Miller et al. model.

The Miller et al. method is based on Bayesian estimation using a $Beta(a,b)$ distribution in which a and b represent prior assumptions based on some information about the software or its development. To continue the efforts of estimating the probability of failure as described by Miller et al., it is necessary to quantify techniques used to improve software during development, testing, and maintenance. The specific problem is using these quantifications to establish the a and b parameters for the $Beta$ distribution.

An extensive search of the literature on white box software analysis techniques is being done. From those models studied, several methods that are believed to give quantitative estimates

concerning probability of failure are being selected. Attempts to model the outcome of these analyses using *Beta* functions will be done. In some cases attempts will be made to experimentally determine whether or not these predictions have statistical validity.

There are three areas already targeted as being a source for reliable quantifiable estimates. Munson and Khoshgoftaar[4] have developed a method of using software metrics to determine fault prone programs. These complexity metrics are statistically analyzed and used to create a predictive model for assigning programs to one of two groups. One group is predicted to have a very low fault rate and the other a very high fault rate. They have devised a method for representing most of the program complexity information as a single value called the relative complexity. A method for using this relative complexity value will be attempted to be derived and used for forming the prior parameters for Bayesian estimation in the Miller et al. model.

Reliability growth models for software have been in use for decades. Applying these models during the development of software can determine predictions of the probability of failure during future tests. From these predictions a mean and a variance can be calculated and used to obtain an a and b to produce a *Beta* distribution having the same mean and variance. Thus, there is a quantifiable source for determining the Bayesian priors.

The Littlewood-Verrall and the Jelinski-Moranda models[2] are well known models that are frequently used in reliability discussions today. Basically, these models make a prediction on the i th version of the software based on the previous 1 through $i-1$ observations of software failures during execution. Data obtained from these models during the development of the software can be used to develop values for a and b .

The model described by Becker and Camarinopoulos[1] is unlike other models discussed in that this model will indicate the possibility that no more errors exist in the software. This method will be analyzed and hopefully incorporated into the determination of the a and b parameters so that the probability of failure equal to zero is represented by the Miller et al. model.

There are several areas currently being considered for further research as far as expanding the techniques for determining the a and b parameters. Continuation of this research will result in several ways for quantifying parameters for Bayesian prior assumptions when estimating the probability of failure of software.

REFERENCES

- [1] G. Becker and L. Camarinopoulos, "A Bayesian estimation method for the failure rate of a possibly correct program," *IEEE Transactions on Software Engineering*, vol 16, pp 1307-1310, November 1990.
- [2] Brocklehurst, P. Y. Chan, B. Littlewood, and J. Snell, "Recalibrating software reliability models," *IEEE Transactions on Software Engineering*, vol 16, pp. 458-469, April 1990.
- [3] K.W. Miller, L. J. Morell, R. E. Noonan, S. K. Park, D. M. Nicol, B. W. Murrill, and J. M. Voas, "Estimating the probability of failure when testing reveals no failures," *IEEE Transactions on Software Engineering*, Vol 18, pp. 33-43, January 1992.
- [4] J. C. Munson and T. M. Khoshgoftaar, "The detection of fault-prone programs," *IEEE Transactions on Software Engineering*, Vol 18, PP 423-432, May 1992.